

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

IN THE SPECIFICATION:

Please replace paragraph [14] beginning at page 8, with the following rewritten paragraph:

[14] Screen markers on the Graphical User Interface (GUI) elements of the Target Software make the Call Agent aware of the GUI elements (e.g., field boxes, buttons, pictures, etc. otherwise known by one schooled in the art as ~~Controls~~ controls) are associated with a pre-defined action (or series of actions) and have been enhanced by the invention. Variations in the screen markers represent different types of automated actions that manipulation of the GUI elements will execute.

Please replace paragraph [23] beginning at page 10, with the following rewritten paragraph:

[23] Figure 7 depicts a ~~FAQsofts~~-FAQsoft Tray Application icon;

Please replace paragraph [50] beginning at page 13, with the following rewritten paragraph:

[50] The PC 10 is used to run Target Applications typically used in Call Centers for taking orders or collecting

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

information from a client. A user interacts with the Target Application's Application's Graphical User Interface (GUI) 30 using a keyboard and/or mouse or other input device. The GUIs may include several different types of windows. At the top level are windows that have title bars and are called Top Level Windows. Other GUI elements hosted by Top Level Windows are commonly referred to as Controls.

Please replace paragraph [53] beginning at page 14, with the following rewritten paragraph:

[53] Data for FAQsoft resides in a FAQsoft Database 27, either in the local hard drive 20 or at a server computer 26. The database stores information mainly on Control/Response association, Phrases, Hotkeys and Users. Information stored for each User includes controls to monitor in the target application, which audio files of recorded phrases to use (only audio files created in that User's User's voice) Audio File directories and that User's User's Hotkeys List, Deskbar settings and User access rights.

Please replace paragraph [55] beginning at page 14, with the following rewritten paragraph:

[55] In an Event based GUIs Environment like Microsoft Windows,

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

application screens are built using various 'windows', such as Edit box, Labels, Frames, Checkboxes, ComboBoxes and etc. These objects are actually windows with different styles and can be contained within other windows. Such windows are also known as 'Controls'. Windows that have a title bar are usually known as Top Level windows. Each window has a function to tell it how to behave when a user interacts with it. When a mouse pointer is clicked on a control, the OS sends a message to that control's windows function. Almost all interactions by the user with application software ~~generates~~ generate one or more messages that are queued within the OS message queue and are forwarded to the respective windows functions. The OS also provide tools (APIs) to allow programmers to trap these messages before they reach the intended windows function.

Please replace paragraph [60] beginning at page 17, with the following rewritten paragraph:

[60] Attention is now directed to Figure 6, which is a flow chart showing the process of invoking and running the FAQsoft mechanism of the present invention. FAQsoft is designed to run silently as a background process and will auto start when the computer is booted up. When FAQsoft is first started at step 100, it installs an icon into the system tray at step 110. System tray icons are icons that reside at the bottom right area of the desktop. They

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

are usually reserved to represent applications that run in the background. Figure 7 depicts the ~~FAQsoft~~ FAQsoft Tray Application icon ~~White~~ White Quotes on ~~Red~~ Red in the system tray after it is installed by this step.

Please replace paragraph [63] beginning at page 18, with the following rewritten paragraph:

[63] To operate FAQsoft, there must always be a valid user logged in (step 140), since it needs to play back personalized voice recordings to the phone system, and therefore must know and prepare the current ~~user's~~ user's audio files. Upon successful login, the current user name is saved in Windows Registry in step 150, and the user's specific information (User Profile) is retrieved and loaded in step 160.

Please replace paragraph [71] beginning at page 20, with the following rewritten paragraph:

[71] Steps 300 - 480 of the routine list the more significant menu items as follows:

- Login step 300 loads the Login Module (step 310) that displays the Login Dialog to change user.
- Phrase Manager step 320 loads the Phrase Manager Module (step 330) that displays the Phrase Manager Window, shown in

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

Figure 11. The Phrase Manager has three main functions:

- 1- Maintenance - including adding, deleting and editing phrase records;
- 2- Recording ~~audio~~ audio recording of phrases; and
- 3- Quality ~~Review~~ Review of recorded phrase by supervisors.

- The Hotkeys Manager step 340 loads the Hotkeys Manager Window shown in Figure 12. The user can drag and drop Response Items from the Configure Manager into this window. The Hotkey Manager will display a dialog as shown in Figure 13 to allow the user to designate a unique key combination that will be used to trigger the playback of a phrase or execution or a Response. The Allow Default Action checkbox will permit pre-existing hotkeys in other applications to be executed concurrently. Double-clicking a Hotkey item shows the Hotkey dialog for further edits. When the save button is click, FAQsoft persists the new Hotkey item and registers the Hotkey.

Please replace paragraph [95] beginning at page 25, with the following rewritten paragraph:

[95] In order to enable the FAQsoft routine to monitor other applications, hooks are installed (step 510) which trap Windows messages sent to these applications. These hooks are implemented as a Windows dll (~~Dynamic~~-Dynamic Link Library)

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

Library) file 12; two separate system wide hooks are installed using the SetWindowsHookEx function call. The Message Hook installed monitors only the WM_ACTIVATE message, in order to monitor focus changes occurring in Top Level Windows only. The Keyboard Hook is installed to implement the Hotkey function. Hotkey combinations found in the Hotkeys List are then registered in step 520. The Keyboard Hook installed traps all registered hotkey messages.

Please replace paragraph [96] beginning at page 25, with the following rewritten paragraph:

[96] By looking at the WM_ACTIVATE message, the FAQsoft mechanism is informed whenever the user changes the active Top Level Window. When a WM_ACTIVATE message is received (step 560), FAQsoft enumerates the Controls contained within the active Top Level Window (step 570) using Win32 API call EnumChildWindows. It tries to match items in the Configure List with Controls that exist within the currently active Top Level Window (step 580). If a match is found, that Control is subclassed (step 590), the hwnd field of the matching items in the Configure List (Configure Items) is updated with the control's window handle (hwnd) (step 600). User interaction with that control is monitored once it has been subclassed.

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

Please replace paragraph [97] beginning at page 26, with the following rewritten paragraph:

[97] With respect to static information, the FAQsoft program extracts embedded information from a control and concatenates them to form a key that uniquely identifies each selected control (Control Static Information). This key is made up of information that remains static between multiple instances of the same Control and is generated in a similar manner as the Spy++ program found in Microsoft's Visual Studio toolset. It includes the ~~Control's~~ Control's Application Name, Class Name, Class ID, Parent Name, Parent ID, Location and a few other specific characteristics of the control obtainable via Win32 API calls that require the window handle (hwnd) value of a control. A Control's Static Information does not include the hwnd of the Control because this value will be different every time the Target Application creates that control.

Please replace paragraph [98] beginning at page 26, with the following rewritten paragraph:

[98] During a previous Configuration process the same key (Static Information) would have been generated when a user selects the

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

Control. This is stored in the Static Information field of the Configure Record. FAQsoft also attempts to supply a descriptive name for that Control and stores it in the Control Name field of the Configure Record. This name is generated differently depending on the selected ~~Control's~~ Control's Class Type. For Controls with a Static Text property, such as Labels, Buttons, Dialog boxes and Icons, their embedded text is used and the control type is appended after the name. For Controls with editable text, the FAQsoft routine finds the closest label situated to the left or top of the selected control, extracts that Label's text value and appends an abbreviated control type description to the Label text to form the Control Name.

Please replace paragraph [99] beginning at page 27, with the following rewritten paragraph:

[99] For example, if the Save button is selected, the FAQsoft routine will use the button name and postfix the control type to the name and the value ~~Save_btn~~ Save_btn will be stored as the Control Name in the Configure Record. As another example, if the Job Title field on a form is selected, the FAQsoft routine will recognize this as an editable text control and proceed to locate the Label for that field and store ~~Job Title_txt~~ Job Title_txt as the Control Name value in its Configure Record. The Configure Record's Control Name field is used as a key principally in

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

search operations called by the User.

Please replace paragraph [104] beginning at page 29, with the following rewritten paragraph:

[104] Once a Control is subclassed, the FAQsoft routine will be aware of user interactions with that control. For example a tab into an edit box will cause the edit box to receive focus. WinOS sends a WM_SETFOCUS message to the edit box which is intercepted (via subclass mechanism) and recognized by FAQsoft as one of the monitored message (step 620). The FAQsoft routine then compares the editbox's editbox's hwnd with hwnds in the Configure List (step 630). If a match is found, the Event parameter is compared to determine a valid match and raises an event (command) for the Tray Application to consume (step 640) as illustrated by the following pseudo-code example:

```
'If hwnd matches then user is manipulating a monitored
control

for each currentHwnd = ConfigItemHwnd
  if currentEvent=ConfigItemEvent then
    currentControlText=read text contents of current control if
    any
    RaiseEvent (currentControlText,ConfigItemResponse)
    Exit For
  end if
```

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

next.

Please replace paragraph [105] beginning at page 29, with the following rewritten paragraph:

[105] The Tray Application (step 650) executes the Response if the Condition parameter is met (step 201). It operates as a separate thread so as not to impede the Monitoring Module's performance. Other Mouse messages received from monitored Controls are treated similarly (step 621).

Please replace paragraph [108] beginning at page 30, with the following rewritten paragraph:

[108] Figure 24 shows a normal Outlook Contact window before running the FAQsoft routine and Figure 25 shows the same window after being configured with FAQsoft and with FAQsoft running. The screen indicators provide a visual cue for the Call Agent to anticipate certain behavior when he or she interacts with FAQsoft's enhanced Controls in the Target application.

Please replace paragraph [113] beginning at page 32, with the following rewritten paragraph:

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

[113] When Configure Mode is enabled in step 810, the user is able to click and drag Phrase and Response items from their respective lists and drop them onto Controls in Target Application windows. Phrase items dropped on controls automatically generates a Response Record with the value Play:-Play: <PhraseName>. When the user clicks on either a Phrase or Response item, a MouseDown event (step 820) on the List (grid Control) that host these items (Figures Figure 14 and Figure 18) triggers a SetCapture API call (step 830), which causes the mouse pointer to be tracked across the screen (step 831). A timer is used to periodically call GetWindowFromPoint() to get the window handle (hwnd) of the Control below the mouse pointer (step 832) and highlight the Control under the mouse.

Please replace paragraph [117] beginning at page 33, with the following rewritten paragraph:

[117] And the user drags in a Response Item called VerifyCardNumber VerifyCardNumber into the same Control, the FAQsoft routine will automatically fill the Condition and Event fields with the following values for the new Configure Item:

```
ControlName = CreditCard #_txt --> CreditCard #_txt
Event = OnEnter --> OnLeave
Condition = Empty --> Not Empty
```

In re Patent Application of:
ALHADDAD ET AL
Serial No. 09/990,871
Filed: NOVEMBER 16, 2001

Response = AskCardNumber --> VerifyCardNumber

Please replace paragraph [122] beginning at page 35, with the following rewritten paragraph:

[122] Upon receiving the event, the FAQsoft routine replaces it with its own Context Menu and displays it (step 892) at the position where the click occurred. The replaced context menu contains only two items, Clear All and Edit. When the user selects Clear All menu item (step 893), all Configure items related to the clicked control will be removed from the Configure List (step 895). The Database is updated and the ~~control's Control~~ subclassed (step 896) is removed. When the user selects Edit menu item, the FAQsoft routine loads the Controls Configure items and displays them in the Configure Info Dialog shown in Figure 27.